상명대학교 컴퓨터과학과 "EA0003: 시스템소프트웨어" 실습

- 실습 번호: lab04

- 실습 디렉터리: ~/lab04 이름: 이경헌

- 실습 날짜: 2017년 5월 24일

- 실습 제목: Array 학번: 201311229

- 주의사항: 최종 실습 파일을 E-campus에 제출할 때 파일 이름은 반드시 lab-03-loop-학번-이름.dox 형태로 할 것.

- 실습 내용:

1. (실습준비) 터미널에서 명령 “mkdir lab04”를 수행하여 자신의 홈 디렉터리 아래에 실습 디렉터리 (~/lab04)를 만든다. 이번 과제에서 만들어지는 모든 파일들은 이 디렉터리에 두도록 한다.
2. (search.asm) 정수 배열에서 원하는 값을 탐색하는 프로그램 search.asm을 작성하시오. 이 프로그램을 어셈블, 링크하여 실행 파일 search를 생성하고, 이를 실행시킨 결과를 화면 캡쳐하여 첨부하시오. 이 프로그램은 다음 조건들을 만족해야 한다.

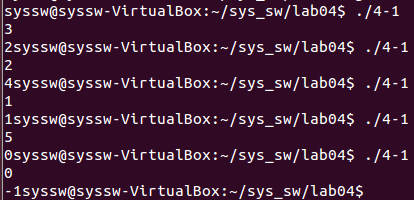
* 정수 배열은 다음과 같이 미리 저장된 배열을 사용

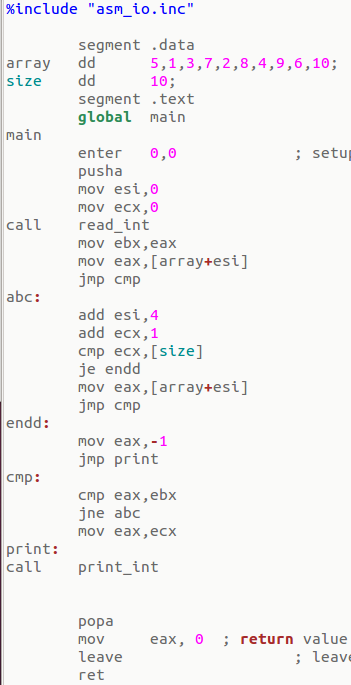
segment .data

array dd 3, 1, 5, 7, 2, 8, 4, 9, 6, 10

size dd 10

* 탐색할 값은 read\_int 루틴을 통해 사용자로부터 입력 받는다.
* 값이 존재하는 경우 해당 인덱스 (첫번째 배열 원소의 인덱스는 0으로 함)를 출력하고, 없으면 -1 을 print\_int 루틴을 사용하여 출력





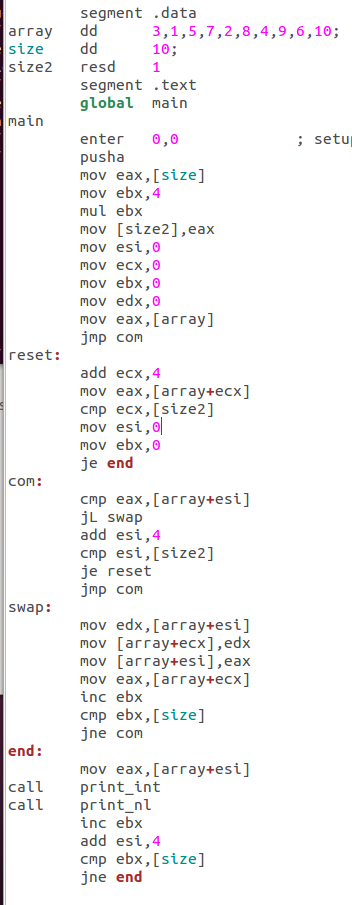
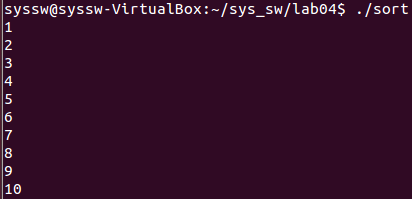
1. (sort.asm) 주어진 정수 배열을 오름차순으로 정렬하는 프로그램 sort.asm 작성하시오. 이 프로그램을 어셈블, 링크하여 실행 파일 sort를 생성하고, 이를 실행시킨 결과를 화면 캡쳐하여 첨부하시오.

* 정렬 대상이 되는 정수 배열은 위의 1번과 동일한 것을 사용.
* 정렬이 완료되면 제대로 정렬 되었는지 확인하기 위해 정렬된 내용을 화면에 출력해 보도록 한다.
* 정렬 방식은 다음의 알고리즘을 사용하거나, 자신이 선호하는 다은 정렬 방법 사용 가능.

void insert\_sort(int list[], int n)  
{  
    int i,j

int next;

    for (i=1; i<n; i++){  
        next=list[i];  
        for (j=i-1; j >=0 && list[j] > next; j--)  
            list[j+1]=list[j];  
        list[j+1]=next;  
    }  
}



1. (mmult.asm) 두 행렬의 곱셈을 수행하는 프로그램 mmult.asm을 작성하시오. 이 프로그램을 어셈블, 링크하여 실행 파일 mmult 을 생성하고, 이를 실행시킨 결과를 화면 캡쳐하여 첨부하시오.

* 곱하는 두 행렬 및 결과값 배열은 다음과 같이 미리 저장된 배열을 사용

segment .data

a dd 1, 2, 3, 4, 5, 6

rows\_a dd 2

cols\_a dd 3

b dd 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9

rows\_b dd 3

cols\_b dd 4

segment .bss

c resd 8

rows\_c resd 1

cols\_c resd 1

* 결과값의 출력은 반드시 다음과 같은 형태로 출력되어야 함

29 35 41 38

74 89 104 83

* 행렬 곱셈 알고리즘은 다음을 참조

for (i = 0; i < rows\_a; i++)

for (j = 0; j < cols\_b; j++) {

sum = 0;

for (k = 0; k < cols\_a; k++)

sum = sum + a[i][k] \* b[k][j];

c[i][j] = sum;

}

끝.

%include "asm\_io.inc"

segment .data

a dd 1, 2, 3, 4, 5, 6

rows\_a dd 2

cols\_a dd 3

b dd 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9

rows\_b dd 3

cols\_b dd 4

segment .bss

c resd 8

rows\_c resd 1

cols\_c resd 1

sum resd 1

segment .text

global main

main

enter 0,0 ; setup stack frame

pusha

mov eax,[rows\_a]

mov [cols\_c],eax

mov eax,[cols\_b]

mov [rows\_c],eax

mov ecx,-4 ; i

mov edx,-4 ; j

mov esi,-4 ; k

mov eax,0

mov [sum],eax

first:

add ecx,4

second:

add edx,4

third:

add esi,4

mov eax,edx

mov ebx,3

mul ebx

mov ebx,[b+eax+esi]

mov eax,ebx

mov eax,[a+2\*esi+ecx]

mul ebx

add [sum],eax

cmp esi,8

jne third

mov eax,[sum]

call print\_int

popa

mov eax, 0 ; return value

leave ; leave stack frame

ret